

Esame - Iscrizione

Assumete le vostre responsabilità!

- **Iscrizione (fino a 3 giorni prima)**
- **De-iscrizione (fino a 24 ore prima, eventualmente via mail. Last minute – only for serious reasons-, via mail.)**

No excuses. No exceptions.

Esame - Materiale

1a prova:

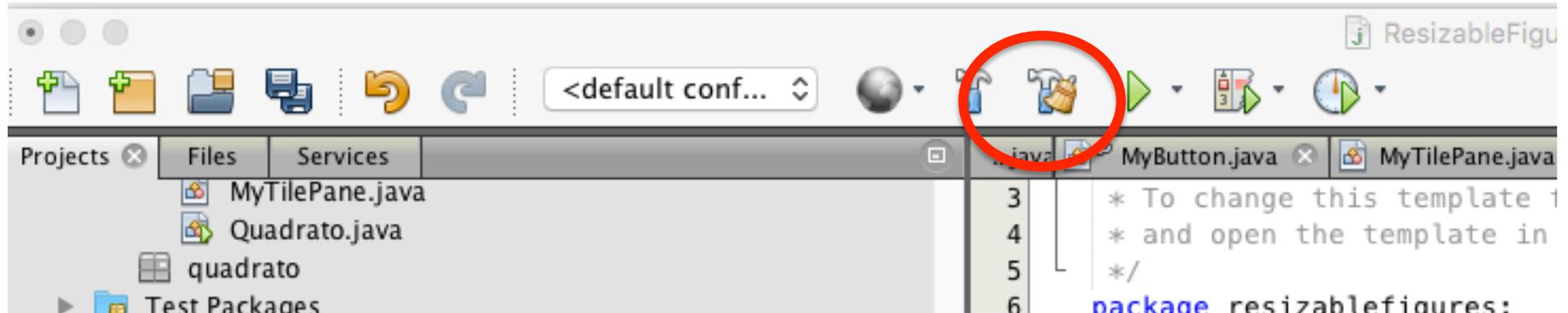
- Orologio
- Penne
- Documento
- N.Matricola

2a prova:

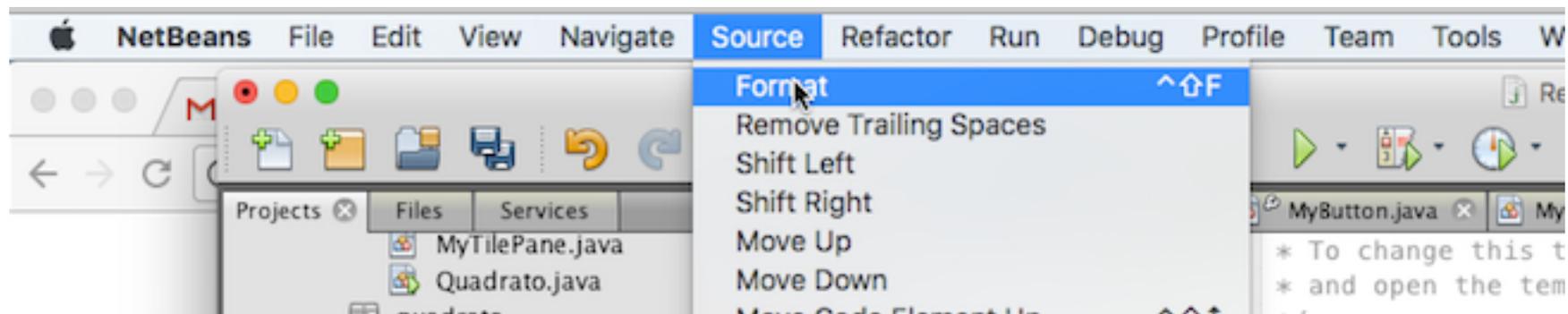
- Tutto il materiale cartaceo che volete – solo per uso personale.
- API in locale, in Netbeans.

Netbeans tricks

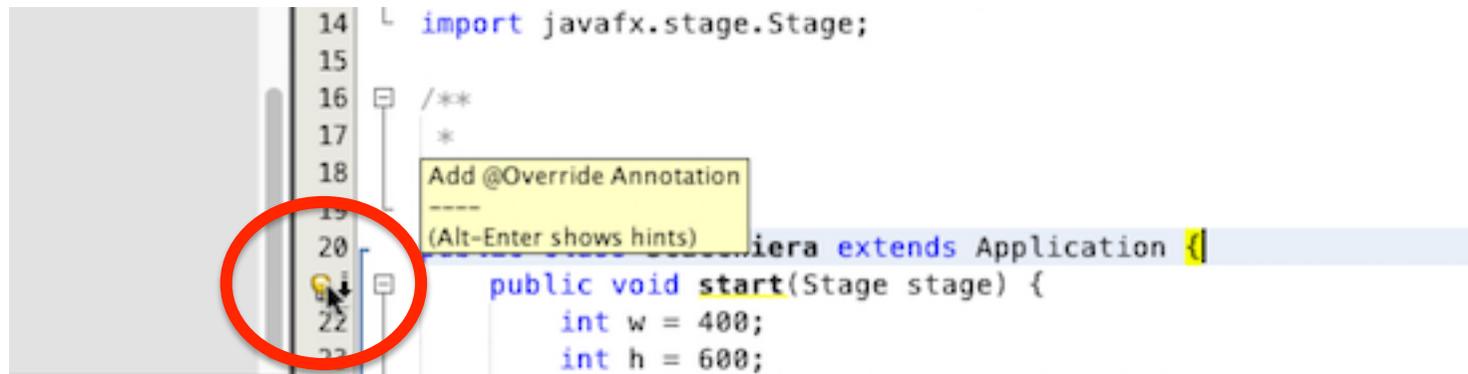
Always fully recompile! (Javafx)



Format!

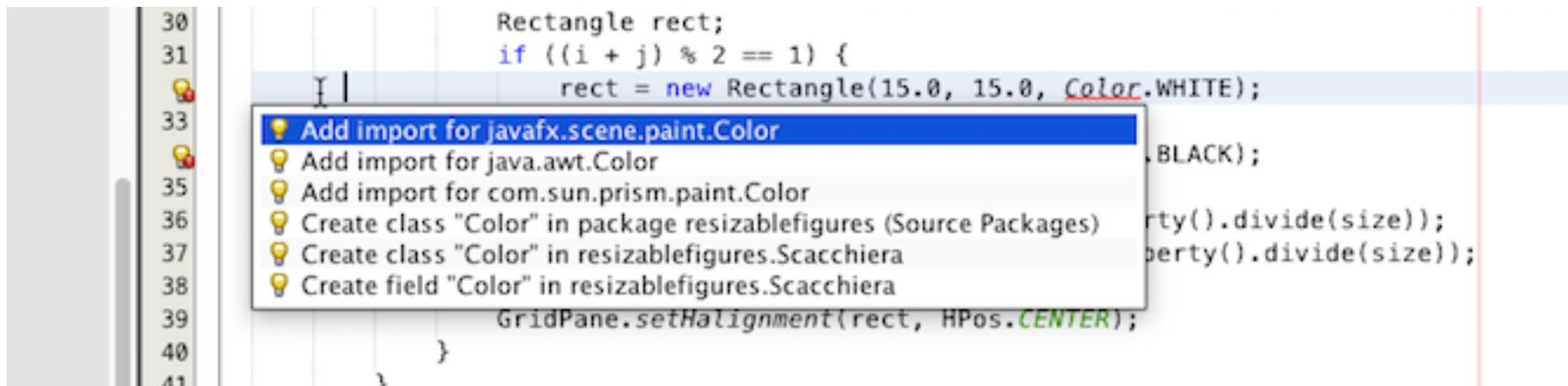


Read the suggestions!



```
14 import javafx.stage.Stage;
15
16 /**
17  *
18  */
19
20 Scacchiera extends Application {
21     public void start(Stage stage) {
22         int w = 400;
23         int h = 600;
```

Be careful with the imports!



```
30 Rectangle rect;
31 if ((i + j) % 2 == 1) {
32     rect = new Rectangle(15.0, 15.0, Color.WHITE);
33 }
34 Color BLACK);
35 }
36 }
37 }
38 }
39 GridPane.setHalignment(rect, HPos.CENTER);
40 }
41 }
```

History

The screenshot shows an IDE interface. At the top, a file explorer shows a project structure with 'ResizableFigures.java' selected. A context menu is open over it, with 'History' and 'Show History' highlighted. Below the explorer is a table of file history. At the bottom, a code diff view compares the current file with a previous version, highlighting a change on line 34.

Date	Message
Yesterday 1:59:26 PM	
Yesterday 1:59:12 PM	
Yesterday 1:58:51 PM	
Yesterday 1:58:24 PM	
Yesterday 1:57:08 PM	
Yesterday 1:54:11 PM	

ResizableFigures.java (Yesterday 1:59:26 PM)	1/1	Current File
<pre>public class ResizableFigures extends Application { @Override public void start(Stage primaryStage) { Rectangle rect = new Rectangle(10, 10, 40, 40); Circle circle=new Circle(30,200,20); Group root = new Group(); final Scene scene = new Scene(root, 300, 250); rect.addEventHandler(MouseEvent.DRAG_DETECTED, new @Override public void handle(Event event) { scene.setCursor(Cursor.HAND); }); } }</pre>		<pre>@Override public void start(Stage primaryStage) { Rectangle rect = new Rectangle(10, 10, 40, 40); Circle circle=new Circle(30,200,20); Group root = new Group(); circle.centerXProperty().bind(rect.widthProperty()); final Scene scene = new Scene(root, 300, 250); rect.addEventHandler(MouseEvent.DRAG_DETECTED, new @Override public void handle(Event event) { scene.setCursor(Cursor.HAND); }); }</pre>

Mouse dragging

Cattura gli eventi di mouseDrag

```
public class ResizableRect extends Application {  
    @Override  
    public void start(Stage primaryStage) {  
        Rectangle rect = new Rectangle(10, 10, 40, 40);  
        Group root = new Group();  
        final Scene scene = new Scene(root, 300, 250);  
        rect.addEventHandler(...  
        root.addEventFilter(...  
        root.getChildren().add(rect);  
        primaryStage.setTitle("Draggable rect");  
        primaryStage.setScene(scene);  
        primaryStage.show();  
    }  
    ...  
}
```



Cattura gli eventi di mouseDrag

```
rect.addEventHandler(MouseEvent.DRAG_DETECTED, new EventHandler() {
    @Override
    public void handle(Event event) {
        scene.setCursor(Cursor.HAND);
    }
});

root.addEventFilter(MouseEvent.MOUSE_RELEASED, new EventHandler() {
    @Override
    public void handle(Event event) {
        scene.setCursor(Cursor.MOVE);
        double x0 = rect.getX();
        double xfinal = ((MouseEvent) event).getX();
        rect.setWidth(xfinal - x0);
    }
});
```



Properties and bindings

Property binding

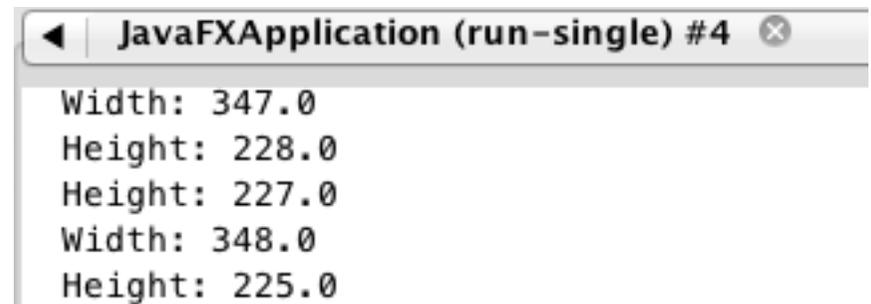
```
Circle circle=new Circle(30,200,20);  
root.add(circle);  
circle.centerXProperty().bind(rect.widthProperty());
```

Property binding

```
int radius=20;
int xcenter=radius+10;
Circle circle = new Circle(xcenter, 200, radius);
Group root = new Group();
rect.widthProperty().addListener(new <Number>() {
    @Override
    public void changed(ObservableValue<? extends Number> ov,
        Number oldWidth, Number newWidth) {
        System.out.println("Width: " + newWidth);
        circle.setCenterX(newWidth.doubleValue()/2+10);
        circle.setRadius(newWidth.intValue()/2);
    }
});
```

Cattura gli eventi di resize della scena

```
public class WindowResize extends Application {
    public void start(Stage stage) {
        Node circ = new Circle(40, 40, 30);
        Parent root = new Group(circ);
        Scene scene = new Scene(root, 400, 300);
        scene.widthProperty().addListener(new ChangeListener<Number>() {
            public void changed(ObservableValue<? extends Number>
                observableVal, Number oldWidth, Number newWidth) {
                System.out.println("Width: " + newWidth);
            }
        });
        scene.heightProperty().addListener(new ChangeListener<Number>() {
            public void changed(ObservableValue<? extends Number>
                observableVal, Number oldHeight, Number newHeight) {
                System.out.println("Height: " + newHeight);
            }
        });
        stage.setScene(scene);
        stage.show();
    }
    ...
}
```



```
JavaFXApplication (run-single) #4
Width: 347.0
Height: 228.0
Height: 227.0
Width: 348.0
Height: 225.0
```

```
public void start(Stage stage) {  
    int w = 400; int h = 500; int size = 8;  
    GridPane board = new GridPane();  
    Scene scene = new Scene(board, w, h);
```

```
    for (int i = 0; i < size; ++i) {  
        for (int j = 0; j < size; ++j) {  
            Rectangle rect;  
            if ((i + j)%2==1)rect = new Rectangle(15.0, 15.0, Color.WHITE);  
            else rect = new Rectangle(15.0, 15.0, Color.BLACK);  
            rect.widthProperty().bind(board.widthProperty().divide(size));  
            rect.heightProperty().bind(board.heightProperty().divide(size));  
            board.add(rect, i, j);  
            GridPane.setHalignment(rect, HPos.CENTER);  
        }  
    }  
    stage.setScene(scene);  
    stage.show();  
}
```

Scacchiera ridimensionabile

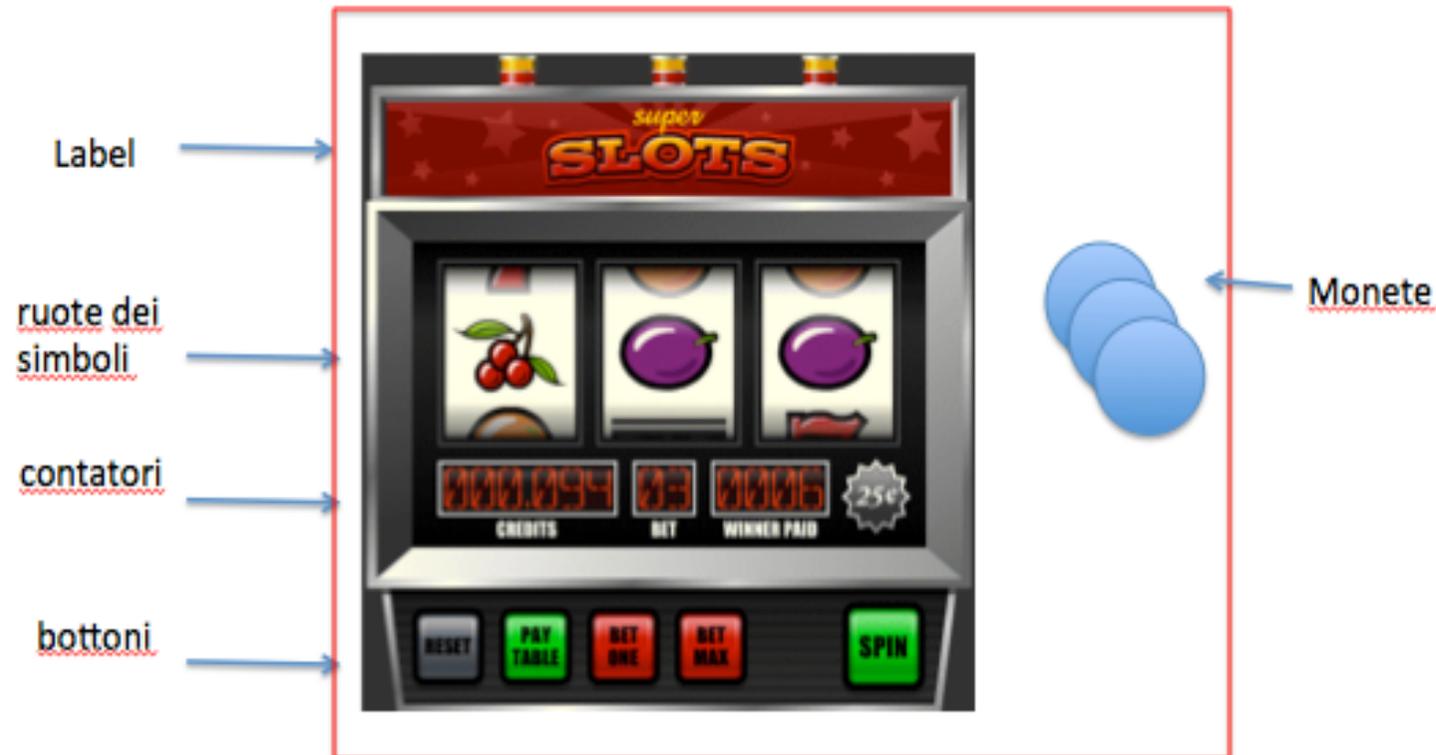
Scacchiera ridimensionabile

```
public void start(Stage stage) {
    int w = 400; int h = 500; int size = 8;
    GridPane board = new GridPane();
    Scene scene = new Scene(board, w, h);
    stage.maxHeightProperty().bind(stage.widthProperty().add(22));
    stage.minHeightProperty().bind(stage.widthProperty().add(22));
    for (int i = 0; i < size; ++i) {
        for (int j = 0; j < size; ++j) {
            Rectangle rect;
            if ((i + j)%2==1)rect = new Rectangle(15.0, 15.0, Color.WHITE);
            else rect = new Rectangle(15.0, 15.0, Color.BLACK);
            rect.widthProperty().bind(board.widthProperty().divide(size));
            rect.heightProperty().bind(board.heightProperty().divide(size));
            board.add(rect, i, j);
            GridPane.setHalignment(rect, HPos.CENTER);
        }
    }
    stage.setScene(scene);
    stage.show();
}
```

Appello Settembre 2014
Anatomia di un'applicazione

Slot Machine

- 1) Scrivere un'applicazione che implementi una slot machine. Tutto il codice deve essere documentato con Javadoc. L'applicazione presenterà una finestra che ricordi vagamente la seguente immagine



2) I contatori sono due:

Credito (indica i soldi disponibili, espressi in centesimi, inizialmente è 0)

Punteggio (inizialmente è 0)

3) Le monete inizialmente sono 3. Sono dei cerchi su ciascuno dei quali è riportata la dicitura "1 Euro".

4) Cliccando su una moneta, questa sparisce e il credito viene aumentato di 100.

5) I bottoni sono :

Nuova partita

Spin (disabilitato se il credito è zero)

Pay (disabilitato se il credito è zero)

6) Le ruote dei simboli sono tre, uguali tra loro. Ciascuna contiene gli stessi sei simboli (delle figure geometriche stilizzate: barra inclinata a destra, rombo, cerchio, ecc., scegliete voi). Ogni ruota mostra un solo simbolo alla volta.

7) Cliccando sul tasto “Nuova partita”, se il credito è inferiore a 100 appare una finestra di pop-up che dice “non hai credito sufficiente”. Altrimenti il credito viene diminuito di 100 e il punteggio viene settato a 128.

8) Se Il tasto Spin è abilitato, cliccandolo i simboli delle tre ruote vengono scelti in modo casuale. Ad ogni pressione del tasto “Spin” il punteggio viene dimezzato (ma se è 1 diventa 0).

9) Cliccando su una delle ruote dei simboli, il suo simbolo viene modificato (ma solo se il punteggio non è zero) scegliendolo in modo casuale (quelli delle altre ruote restano immutati). Il punteggio viene dimezzato.

10) Se i simboli mostrati dalle tre ruote sono uguali, appare una finestra di pop-up che dice “Hai vinto”, il credito viene incrementato di un valore pari al punteggio moltiplicato per 100, il punteggio diventa zero.

11) Cliccando sul tasto “Pay” appare un pop-up che dice “Hai vinto XX Euro”, dove XX è il credito diviso 100. Il sistema viene resettato nella condizione iniziale.

12) Al punto 4, la sparizione della moneta avviene con una traslazione che la fa arrivare sopra la slot machine. La traslazione dura un secondo.

13) Ai punti 8 e 9, il cambiamento di simbolo avviene con il dissolvimento del simbolo “vecchio” e l’apparizione del simbolo “nuovo”. La transizione dura 1 secondo.